Bellcomm

955 L'Enfant Plaza North, S.W.
Washington, D. C. 20024

B71 07045

date: July 27, 1971

to: Distribution

from: I. Y. Bar-Itzhack

subject: A Description of the Rover Navigation System Simulation Program - Case 320

MEMORANDUM FOR FILE

1.0 INTRODUCTION

The simulation program described in this memorandum was written in order to supply a tool by which the performance of the Lunar Roving Vehicle (LRV) navigation (NAV) system could be evaluated. In particular, one is interested in evaluating the error in the NAV system resulting from gyro drift rates, gyro misalignments, odometer errors and errors associated with different traverses, lurains and LRV speeds.

Two parallel NAV systems are simulated in this program, the first of which is the accurate or reference system. This simulated system has no measurement errors and it performs a most accurate computation in order to determine the LRV position based on the gyro and odometer readings. The second system is the real (or computed) system which simulates the actual LRV NAV system. This system uses erroneous measurements due to wheel slip, gyro misalignment, gyro drift and intertial rotation on the moon. This system also assumes erroneously that the heading reading of the gyro is the angle between the projection of the LRV advance vector on the North-East plane and North. The comparison between the reference system and the real system indicates the accuracy of the LRV NAV system.

A schematic diagram of the simulation program is shown in Figure 1. As an input to both the systems discussed here, we may assume a certain deterministic traverse (box 3) or a random traverse (boxes 1 and 2) or both. The deterministic traverse contains inputs to the NAV systems which simulate a superposition of sinusoidal, three-dimensional angular motions which are added to the gross traverse. The gross traverse is a low frequency or a constant angular motion which describes the angular time history of the LRV while performing a motion along the gross details of the traverse. The generation of the random LRV attitude occurs in two stages. At the first stage, after the selection of LRV speed and one of four kinds of lurains, the program generates four signals which simulate the elevation of the four LRV wheels as a function of time. These elevations comply with the given Power Spectrum Density (PSD) curve for the selected speed and lurain and with certain logical assumptions on the Cross Power

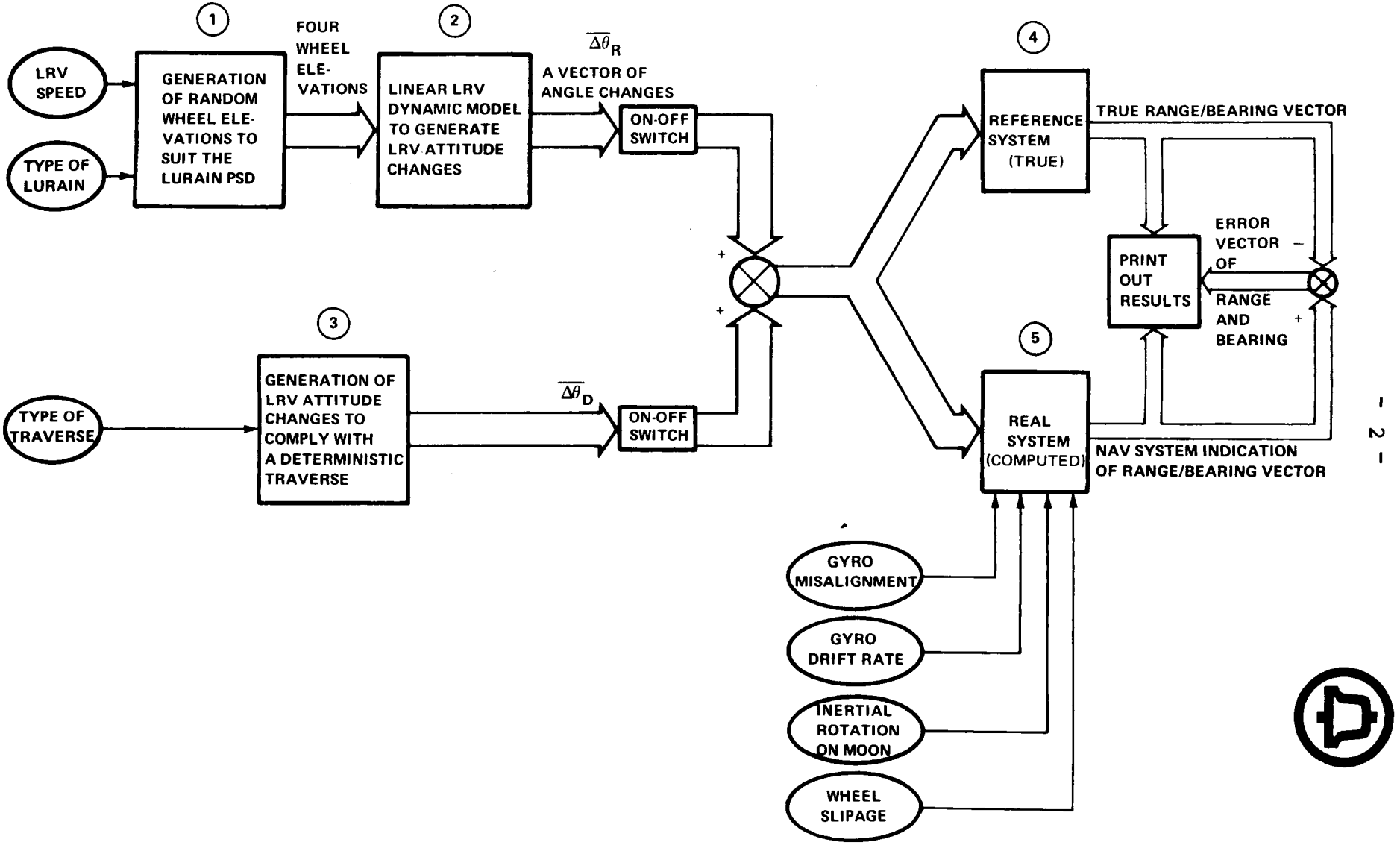


FIGURE 1 - SCHEMATIC DIAGRAM OF THE SIMULATION PROGRAM OF THE LUNAR ROVER NAVIGATION SYSTEM

Spectrum Density (CPSD) of the wheels. This operation is represented by box 1 in Figure 1. The four wheel elevations generated in this stage feed into the linear LRV dynamic model (box 2) from which the resultant LRV angular motions are computed. The outputs of boxes 2 and 3 are three angle differences between the present orientation of the LRV coordinates and this coordinates orientation as it was one iteration ago.

The angular information enters boxes 4 and 5. In box 4 a most accurate computation of the LRV position is being performed. This is done by, first, computing the new Euler angles which describe the LRV attitude after the change of the LRV orientation. Secondly the three dimensional vector which describes the advance of the LRV on the moon during the last iteration interval is projected on the local lunar plane defined by the North-East directions. This projected component is resolved into North and East components which are added respectively to the North and East registers. Finally from the contents of these registers the range from the LM and bearing to the LM is computed. In box 5, the simulation of the real NAV system is carried on. The vector of angle differences which feeds box 4 is contaminated here with erroneous gyro readings due to gyro drift and due to the fact that the gyro, being an inertial measuring device, measures the rotation of the LRV w.r.t. (with respect to) an inertial system although it is interpreted as a rotation w.r.t. the lunar surface. The Euler angles corresponding to the real NAV system are computed in this box taking in account gyro misalignment. The last computation yields the heading that the real gyro would have measured. The Central Processing Unit (CPU) of the real NAV system erroneously interprets this reading as the azimuth of LRV as projected on the North-East lunar plane corresponding to the LRV computed position and hence resolves the LRV advance, which contains wheel slipage, into North and East components which are added to the North and East registers of the simulated true LRV NAV system. As was the case for the perfect system, the indicated range and bearing are also computed from the contents of the north and east registers. Periodically, the program prints out the correct and the computed ranges and bearings and their differences which indicate the errors of the simulated real LRV NAV system. The basic units system used in this simulation is the MKS system. Sometimes the time is converted to minutes for clear print-out reasons.

2.0 DESCRIPTION OF THE MAIN PROGRAM (LRVNAV)

A Flow Chart describing the sequences of the operations executed in the main program is shown in Fig. 2. To facilitate the correspondence between this write-up and the listing of the main program, we will divide the description of

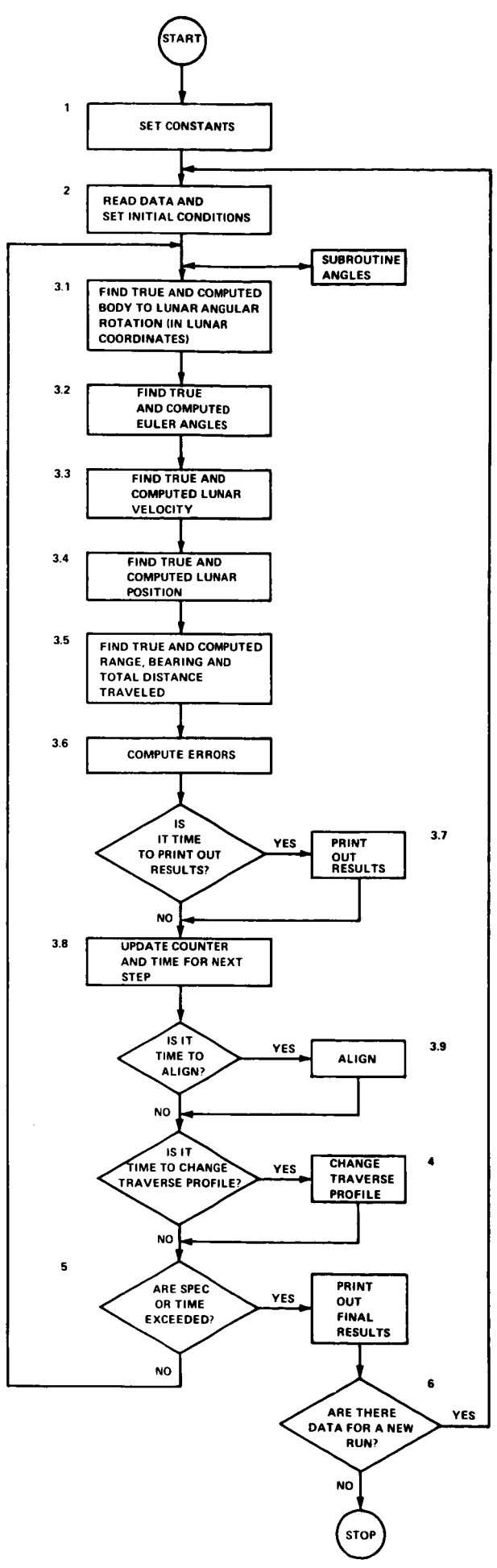


FIGURE 2 A FLOW CHART FOR LHVNAV ROUTINE

the main routine according to the division of the listing with the difference that here we start the numbering of the paragraphs with the number 2.

The program, which is designated as LRVNAV, starts with real, double precision, dimension, common and data statements and then it proceeds as follows.

2.1 Constants

The program sets up the values of the constants used in this program such as lunar rotation rate, lunar radius, etc.

2.2 Initial Conditions

The set of initial conditions for the present run are read from data cards, while the initial conditions which are common to all the runs are reset in the program itself.

2.3 Start Simulation Cycle

The LRV is assumed to be in motion and the simulation of the LRV NAV system starts.

2.3.1 Find True and Computed Body to Lunar Angular Rotation (In Lunar Coordinates)

The main program calls subroutine ANGLES which generates three angle differences. They are the pitch, roll and yaw angular incremental changes which occurred between the last NAV simulation time and the present NAV simulation time. These increments are given in the local lunar coordinate system as shown in Figure 3.

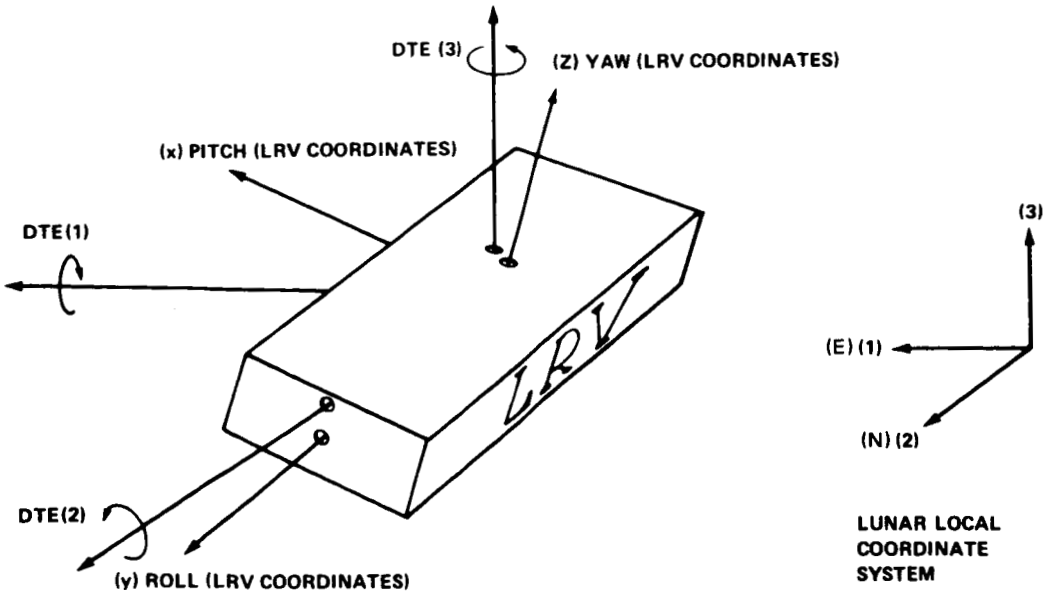
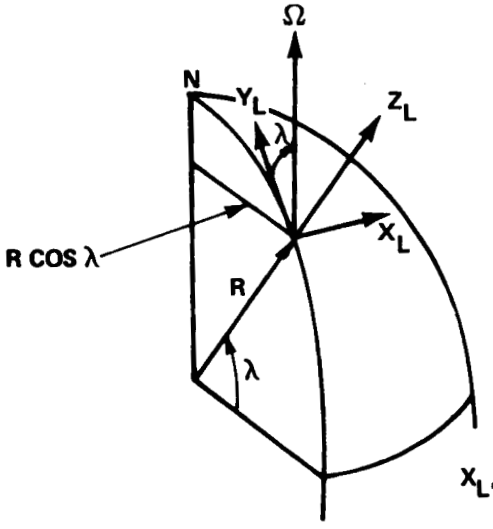


FIGURE 3 - LRV ALTITUDE ANGLE INCREMENTS RESOLVED ALONG THE LOCAL LUNAR COORDINATE SYSTEM

After being generated, these increments are divided by the time increment through which they occurred to create, WA, the LRV angular velocity w.r.t. the lunar surface which is resolved along the local lunar coordinate system. These angular velocities are the entries to boxes 4 and 5 of Figure 1. In box 5, the processed angular velocities are the inertial ones because the gyro measures inertial rotations. So, in addition to these angular velocities, the gyro also measures the angular velocities of the local lunar coordinate system w.r.t. an inertial frame and therefore the two kinds of angular velocities are added to yield the angular velocity vector (WAC) processed in box 5, the simulated real LRV NAV system. From Figure 4 it is seen that ω_{LI}, the angular velocity vector of the local lunar coordinate system with respect to an inertial one, resolved along the lunar coordinates, is given by

$$\omega_{LI} = \begin{bmatrix} -\frac{\dot{Y}_L}{R} \\ \left(\frac{\dot{X}_L}{R \cdot \cos\lambda} + \Omega\right) \cos\lambda \\ \left(\frac{\dot{X}_L}{R \cdot \cos\lambda} + \Omega\right) \sin\lambda \end{bmatrix} \qquad (1)$$

where R is the radius of the moon, λ is the latitude of the LRV, Ω is the moon rotation rate and $\dot{X}_L$ and $\dot{Y}_L$ are, respectively, the East and North velocities of the LRV w.r.t. the moon ($\dot{\lambda}$ is small enough to be neglected).

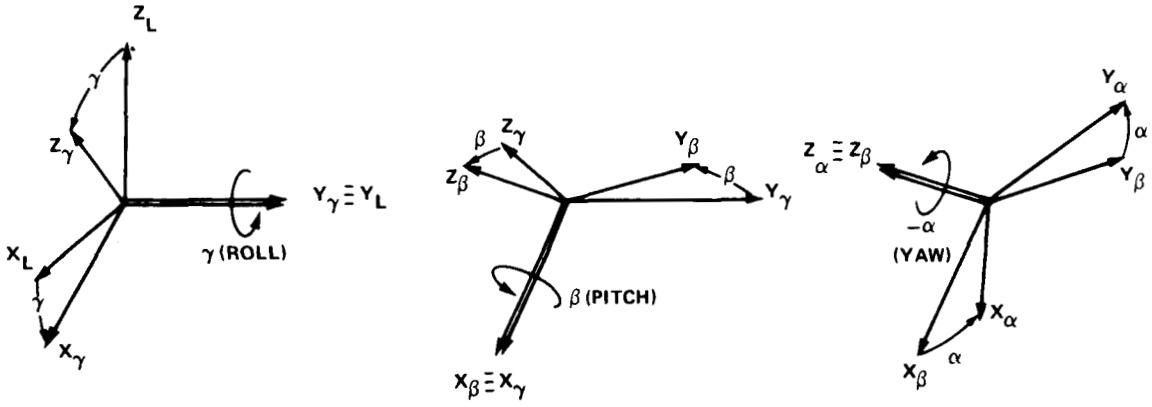


X_L, Y_L, Z_L ARE THE AXES OF THE LOCAL LUNAR COORDINATE SYSTEM

FIGURE 4 - THE ROTATION OF THE LOCAL LUNAR COORDINATE SYSTEM

2.3.2 Find True and Computed Euler Angle

Next the program calculates the Euler angles in box 4 as well as in box 5. No matter how the perfect model of the LRV has rotated up to the present time, it can always be considered as a successive rotation about three axes in space. We choose the sequence as shown in Figure 5 because the gyro reading is one of the Euler angles rather than being a non-linear function of the two or the three of them. The reading of the directional gyro (DG) is the Euler angle α.



(a) ROTATION OF THE L (LOCAL LUNAR) SYSTEM BY AN ANGLE γ ABOUT THE X_L AXIS TO THE γ SYSTEM

(b) ROTATION OF THE γ SYSTEM BY AN ANGLE β ABOUT THE X_γ AXIS TO THE β SYSTEM

(c) ROTATION OF THE β SYSTEM BY AN ANGLE $-\alpha$ ABOUT THE Z_β AXIS TO THE α SYSTEM. THE α SYSTEM IS THE LRV SYSTEM!

FIGURE 5 - DEFINITION OF EULER ANGLES FOR THE LRV ORIENTATION WITH RESPECT TO THE LOCAL LUNAR COORDINATE SYSTEM

Going back to Figure 2, we see that between the previous and present updates of the system, the orientation, and hence the Euler angles of the vehicle, have changed due to the existence of WA, an angular velocity between the LRV and the lunar surface as defined in the previous paragraph. In order to compute the formulae for updating the Euler angles, one should consider the relatinship between the angular velocity WA and the rate of change of the Euler angles. Let D_L^γ be the direction cosine matrix which transforms vectors from the γ to the L coordinate system (Figure 5.a) and D_γ^β from the β to the γ coordinate system (Figure 5.b). It can easily be shown that:

$$D_L^\gamma = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix} \cdot\cdot\cdot(2) \quad \text{and} \quad D_\gamma^\beta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & \sin\beta \\ 0 & -\sin\beta & \cos\beta \end{bmatrix}.$$

It is obvious then that the direction cosine matrix D_L^β which transforms vectors from the β to the L coordinate system is the multiplication of these two matrices; that is,

$$D_L^\beta = D_L^\gamma D_\gamma^\beta = \begin{bmatrix} \cos\gamma & \sin\beta\cdot\sin\gamma & \sin\gamma\cdot\cos\beta \\ 0 & \cos\beta & -\sin\beta \\ -\sin\gamma & \sin\beta\cdot\cos\gamma & \cos\beta\cdot\cos\gamma \end{bmatrix}. \qquad (3)$$

From Figure 5.a we realize that $\dot{\gamma}$ is a vector along the Y_L axis in the L coordinate system; therefore, in this system its contribution to the rotation rate of the LRV is given by the vector

$$\begin{bmatrix} 0 \\ \dot{\gamma} \\ 0 \end{bmatrix}. \qquad (4)$$

$\dot{\beta}$ is directed along the X_γ axis in the γ system; therefore, its contribution to the LRV given in the L coordinate system is

$$D_L^\gamma \begin{bmatrix} \dot{\beta} \\ 0 \\ 0 \end{bmatrix}. \qquad (5)$$

Similarly the contribution of $-\dot{\alpha}$ is

$$D_L^\beta \begin{bmatrix} 0 \\ 0 \\ -\dot{\alpha} \end{bmatrix}. \qquad (6)$$

The sum of equations (4), (5) and (6) yields the angular velocity of the LRV w.r.t. the lunar surface given in the local lunar

coordinates which by our previous definition is the vector WA as defined in the previous section; hence:

$$\begin{bmatrix} WA(1) \\ WA(2) \\ WA(3) \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\gamma} \\ 0 \end{bmatrix} + D_L^{\gamma} \begin{bmatrix} \dot{\beta} \\ 0 \\ 0 \end{bmatrix} + D_L^{\beta} \begin{bmatrix} 0 \\ 0 \\ -\dot{\alpha} \end{bmatrix} . \qquad (7)$$

Using equations (2) and (3) in equation (7) yields

$$\begin{bmatrix} WA(1) \\ WA(2) \\ WA(3) \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\gamma} \\ 0 \end{bmatrix} + \begin{bmatrix} \cos\gamma \\ 0 \\ -\sin\gamma \end{bmatrix} \dot{\beta} + \begin{bmatrix} -\sin\gamma \cdot \cos\beta \\ \sin\beta \\ -\cos\beta \cdot \cos\gamma \end{bmatrix} \dot{\alpha} ,$$

which can be written in a matrix form as

$$\begin{bmatrix} WA(1) \\ WA(2) \\ WA(3) \end{bmatrix} = \begin{bmatrix} 0 & \cos\gamma & -\sin\gamma \cdot \cos\beta \\ 1 & 0 & \sin\beta \\ 0 & -\sin\gamma & -\cos\beta \cdot \cos\gamma \end{bmatrix} \begin{bmatrix} \dot{\gamma} \\ \dot{\beta} \\ \dot{\alpha} \end{bmatrix} . \qquad (8)$$

Equation (8) can be easily solved for the Euler angle rates $\dot{\alpha}$, $\dot{\beta}$ and $\dot{\gamma}$ in terms of the vector WA and the Euler angles. When the solutions are integrated we get the increments of the Euler angles accumulated during the simulation time increment (DT). It can be easily shown that they are:

$$\Delta\gamma = \int_T^{T+DT} [\tan\beta \cdot \sin\gamma \cdot WA(1) + WA(2) + \tan\beta \cdot \cos\gamma \cdot WA(3)]dt \qquad (9a)$$

$$\Delta\beta = \int_T^{T+DT} [\cos\gamma \cdot WA(1) - \sin\gamma \cdot WA(3)]dt \qquad (9b)$$

$$\Delta\alpha = -\int_T^{T+DT} \frac{1}{\cos\beta} [\sin\gamma \cdot WA(1) + \cos\gamma \cdot WA(3)]dt . \qquad (9c)$$

Equations (9) are implemented in the program. The integration method used is the simple trapezoid rule which was found to be

sufficiently accurate. After $\Delta\gamma$, $\Delta\beta$ and $\Delta\alpha$ are computed, they are added to γ, β, α respectively.

As one may suspect, a case of an analytic gimbal lock may occur here since Euler angles are being used. From equations (9) it is obvious that gimbal lock occurs when $\cos\beta = 0$ or $\beta = \pm \frac{\pi}{2}$. This however means that the LRV has overturned and we exclude such a case from this simulation.

The computation of the Euler angles as described here is done for the perfect NAV system. There are, however, some changes in the computation of the Euler angles of the real NAV system. (In this program, symbols ending with the letter C denote that they are symbols belonging to the real system.) First, the entry is not the vector WA but rather the vector WAC whose generation was described in the preceeding section. Secondly, the initial Euler angles are different than those of the perfect system, which are set to zero, because of gyro misalignment, thus they are the gyro misalignment errors expressed in Euler angles. Finally, the gyro drift rate has to be added to the expresssions obtained for $\dot{\gamma}_c$, $\dot{\beta}_c$ and $\dot{\alpha}_c$. It can be seen however, that the directional gyro drifts only in the yaw angle. In conclusion, the equations of the true system, parallel to equations (9) of the perfect system, are

$$\Delta\gamma_c = \int_T^{T+DT} [\tan\beta_c \cdot \sin\gamma_c \cdot WAC(1) + WAC(2) + \tan\beta_c \cdot \cos\gamma_c \cdot WAC(3)]dt$$

$$\Delta\beta_c = \int_T^{T+DT} [\cos\gamma_c \cdot WAC(1) - \sin\gamma_c \cdot WAC(3)]dt$$

$$\Delta\alpha_c = -\int_T^{T+DT} \left\{ \frac{1}{\cos\beta_c} [\sin\gamma_c \cdot WAC(1) + \cos\gamma_c \cdot WAC(3)] - YDR_r \right\} dt$$

where YDR_r is the yaw drift rate given in radians per second.

2.3.3 Find True and Computed Lunar Velocity

Given V, the velocity of the LRV, one can find the distance covered by the LRV during the simulation time interval. We assume that this time interval is short enough, relatively

to the LRV time constants, such that the LRV speed and orientation do not vary during this time interval. Now it is possible to project V on the local lunar plane (East-North plane) and resolve the projection into East (x) and North (y) components. Going back to Figure 5, one can find the direction cosine matrix which transforms vectors from the local lunar to the LRV coordinate system. (An easy way to find it is to use Piograms.[1,2]) The transformation of the local lunar (L) coordinates to the LRV coordinates (α) is given by:

$$\begin{bmatrix} X_\alpha \\ Y_\alpha \\ Z_\alpha \end{bmatrix} = \begin{bmatrix} \cos\alpha\cdot\cos\gamma - \sin\alpha\cdot\sin\beta\cdot\sin\gamma & -\sin\alpha\cdot\cos\beta & -\cos\alpha\cdot\sin\gamma - \sin\alpha\cdot\sin\beta\cdot\cos\gamma \\ \sin\alpha\cdot\cos\gamma + \cos\alpha\cdot\sin\beta\cdot\sin\gamma & \cos\alpha\cdot\cos\beta & -\sin\alpha\cdot\sin\gamma + \cos\alpha\cdot\sin\beta\cdot\cos\gamma \\ \cos\beta\cdot\sin\gamma & -\sin\beta & \cos\beta\cdot\cos\gamma \end{bmatrix} \begin{bmatrix} X_L \\ Y_L \\ Z_L \end{bmatrix} \qquad (10)$$

We chose the LRV roll axis to be the y axis of the LRV coordinate system (Figure 3), that is, V is directed along the y_α axis. We also chose X_L axis and the Y_L axis to be the local lunar East and North respectively (Figure 3). From equation (10) we see that

$$\vec{V} = |V|\cdot\vec{y}_\alpha = |V|[(\sin\alpha\cdot\cos\gamma+\cos\alpha\cdot\sin\beta\cdot\sin\gamma)\vec{X}_L + \cos\alpha\cdot\cos\beta\vec{Y}_L + (\cdots)\vec{Z}_L] \quad . \qquad (11)$$

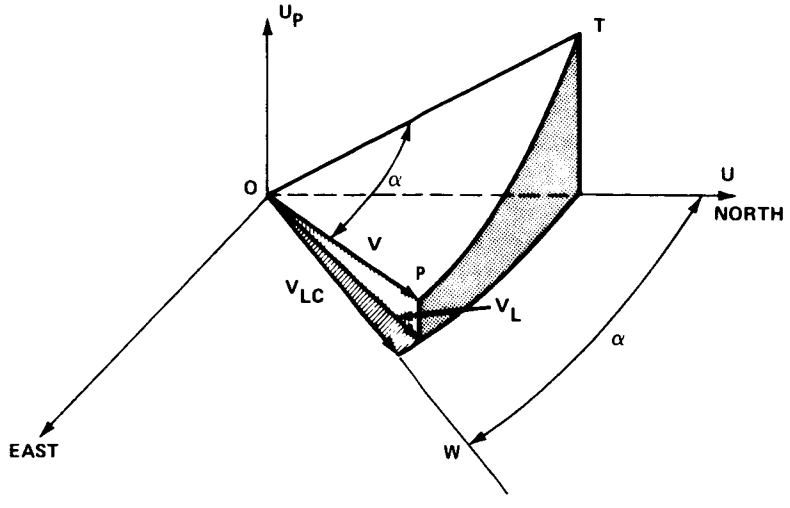


FIGURE 6 - THE EFFECT OF YAW ANGLE MEASUREMENT IN A TILTED PLANE

This means that the true East-ward velocity of the LRV during the simulation interval is

$$V_e = |V|(\sin\alpha\cdot\cos\gamma+\cos\alpha\cdot\sin\beta\cdot\sin\gamma) \quad (12)$$

and the North-ward velocity is

$$V_n = |V|\cdot\cos\alpha\cdot\cos\beta \quad (13)$$

To see how the corresponding computation is done for the real system let us temporarily assume that the real system measurements are accurate and let us consider Figure 6. Between two time increments the LRV has advanced by the distance Δs from point O to point P. The yaw angle α which is measured by the gyro is an angle in the plane TOP which is not parallel to the East-North plane. In the perfect system we take this fact in account and therefore we use equations (12) and (13) which means that we realized that the projection of V on the East-North plane is V_L in Figure 6. In the real system, however, the angle α is assumed to be measured in the East-North plane, that is, $\alpha = \measuredangle$ UOW and therefore the projection of V on the East-North plane is taken as V_{Lc} in Figure 6 rather than V_L. Therefore, rather than using equations (12) and (13), the real system uses the following equations

$$V_{ec} = |V|\cdot\sin\alpha \quad (14)$$

$$V_{nc} = |V|\cdot\cos\alpha \quad (15)$$

and, indeed, when $\beta = \gamma = 0$, that is, when α is really measured in a plane parallel to the East-North plane, we see that equations (12) and (13) are reduced to equations (14) and (15). Actually the real system measures α_c and V_c, then added to the wrong algorithm for computing V_{ec} and V_{nc} we get

$$V_{ec} = |V_c|\cdot\sin\alpha_c \quad (16)$$

$$V_{nc} = |V_c|\cdot\cos\alpha_c \quad . \quad (17)$$

2.3.4 Find True and Computed Lunar Positions

The trapezoid integration rule is used here on equations (12), (13), (16) and (17) to yield the perfect (true) and

the real (computed) lunar positions. (Note that the newly computed true Y (North) position is used in the next simulation pass to find the new latitude angle.)

2.3.5 Find True and Computed Range, Bearing and Total Distance Traveled

The range of the reference LRV system to the LM (Lunar Module) as well as the range of the true system is found by taking the square root of the sum of the squares of the East and North components of the distance. The bearing of the two systems to the LM is found by computing the arctan in degrees of the East components divided by the respective North components of the position and adding 180 degrees to the results. The total distance covered (on the three dimensional traverse) is found by

$$D = \sum_i V_i \cdot DT \quad .$$

2.3.6 Compute Errors

In this section, the operation symbolized by the summation junction on the right hand side of Figure 1 is executed. The error in x (East), the error in y (North), the range and bearing errors are found by subtracting these quantities as computed by the reference system from the corresponding quantities as computed by the real system.

2.3.7 Print Results

At this section, the values of the significant variables of the simulation are printed. The times of the print-outs are optional and determined by input data.

2.3.8 Update Counter and Time for Next Step

At this point, the program adds 1 to the counter which counts the number of passes through the simulation loop (not counting the pass at zero time, that is, when T = 0). The program also updates the time (T) for the next pass.

2.3.9 Align at the Right Time

When the simulation time reaches the predetermined time for alignment, the heading of the real system is set to that of the perfect system plus the misalignment error; that is, the program executes the following equation:

$$\alpha_c = \alpha + YM$$

where YM is the yaw misalignment error.

2.4 Change Traverse Profile

In this section the program checks whether it is time to change to a new traverse leg (input data). If it is, the program picks the suitable LRV speed, type of lurain, slip, initial azimuth of the new leg, rate of azimuth change on this leg and the time the LRV is parked before the beginning of the leg. The program adds the drift accumulated by the gyro during this parking time to the gyro reading α_c unless a realignment is carried out at this time point. (Subroutine PARKLR is used to compute the added drift due to lunar rotation.)

2.5 Stop if Spec is Exceeded and Check for Terminal Time

At this point, the program checks whether to continue the simulation or to exit the simulation loop. If the indicated maximum traverse time or the indicated maximum traverse distance (both specified in the input data) are reached, the program exits the simulation loop. The same also happens if the simulated LRV NAV system errors reach the maximum allowable errors specified in the input data.

2.6 Check for New Run

When the program exits the simulation loop, it reaches this point where it plots the actual traverse on a 4020 plotter and checks the data for an indication to start a new run. If the result is negative, the program stops. If the result is positive, the program starts a new traverse simulation.

3.0 THE GENERATION OF TRAVERSE RELATED ANGULAR INCREMENTS

It was mentioned in section 2.3.1 that at the beginning of the simulation cycle the main program, LRVNAV, calls subroutine ANGLES which generates three angular differences. They are the pitch, roll and yaw angular incremental changes which occurred between the previous NAV simulation time and the present one and are given in the local lunar coordinate system. In this subroutine a linear simulation of the LRV dynamics is carried out where the excitations of the system are four random signals which simulate the elevation of the LRV wheels. In other words, here the motion of the LRV is simulated in order to obtain information on the instantaneous orientation of the LRV NAV system. The equations of the linear LRV dynamic model are per Stanley Kaufman of Bellcomm.

Consider the model shown in Figure 7

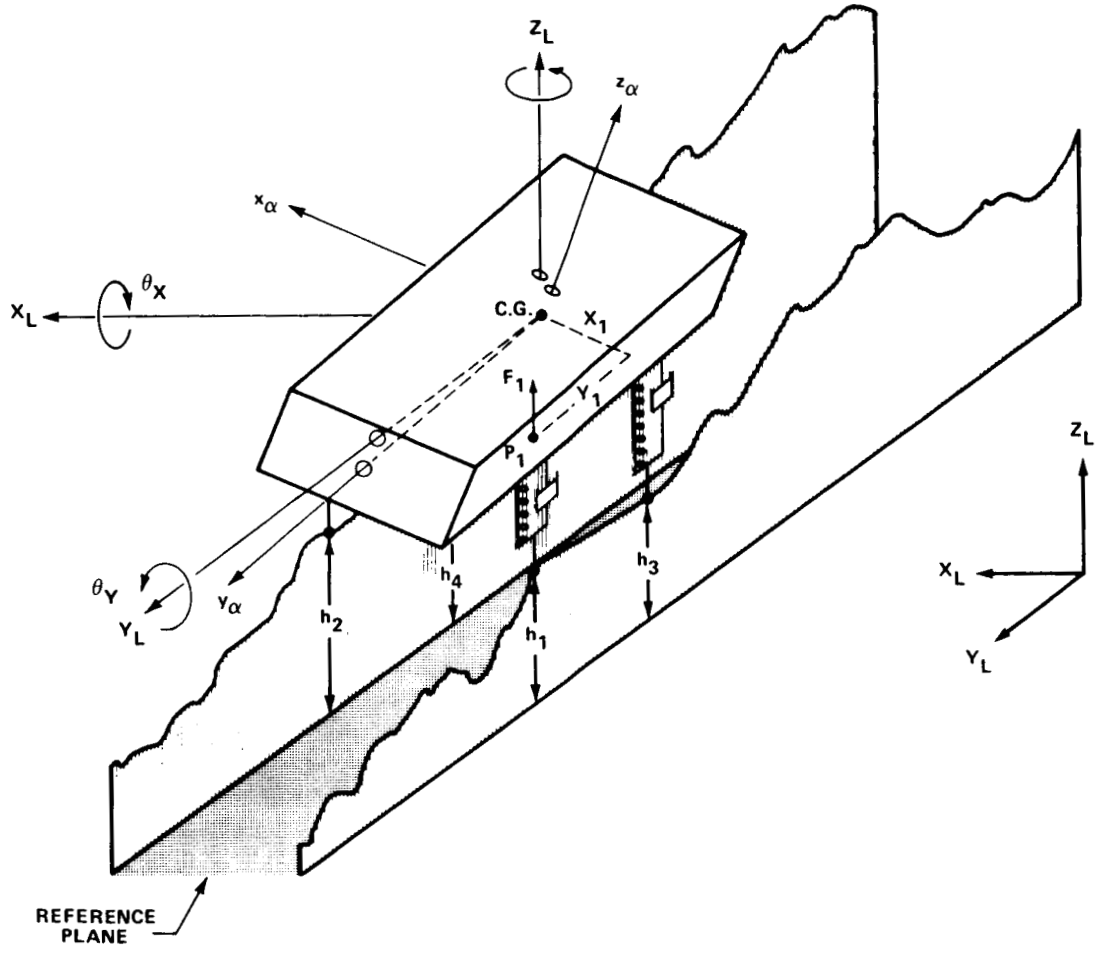


FIGURE 7 - LRV MODEL FOR DYNAMICS ANALYSIS

The force F_1 exerted by the lunar surface at wheel 1 is given by

$$F_1 = K_1(h_1-Z_1) + B_1(\dot{h}_1-\dot{Z}_1) + f_{10}$$

where K_1 and B_1 are the suspension spring constant and the viscous friction coefficient respectively, Z_1 is measured from the reference location of point p_1 when h_1 is zero and the LRV is resting parallel to the reference plan. f_{10} is the value of F_1 when the LRV is at this resting position. This equation is true for all the four wheels, hence we can write:

$$F_i = K_i(h_i - Z_i) + B_i(\dot{h}_i - \dot{Z}_i) + f_{i0} \qquad i=1,2,3,4 \quad .$$

For small angles we can write

$$Z_i \simeq Z_0 - x_1\theta_y + y_1\theta_x$$

where Z_0 is the elevation of the LRV c.g. above its location when the h_i's are zero and the LRV is resting parallel to the reference plane, and x_1 and y_1 are the coordinates of P_1 in the $X_\alpha Y_\alpha$ plane. It can be shown that the last equation holds for all the four wheels, that is

$$Z_i \simeq Z_0 - x_i\theta_y + y_i\theta_x \qquad i=1,2,3,4 \quad . \tag{19}$$

Substituting equation (19) in (18) yields

$$F_i = K_i(h_i - Z_0 + x_i\theta_y - y_i\theta_x) + B_i(\dot{h}_i - \dot{Z}_0 + x_i\dot{\theta}_y - y_i\dot{\theta}_x) + f_{i0} \tag{20}$$

$$i=1,2,3,4 \quad .$$

Therefore the total force on the LRV in the Z_L direction is given by

$$\begin{aligned} F = \sum_{i=1}^{4} F_i = &-\left(\sum_{i=1}^{4} K_i\right)Z_0 + \left(\sum_{i=1}^{4} K_i x_i\right)\theta_y - \left(\sum_{i=1}^{4} K_i y_i\right)\theta_x + \sum_{i=1}^{4} K_i h_i \\ &-\left(\sum_{i=1}^{4} B_i\right)\dot{Z}_0 + \left(\sum_{i=1}^{4} B_i x_i\right)\dot{\theta}_y - \left(\sum_{i=1}^{4} B_i y_i\right)\dot{\theta}_x + \sum_{i=1}^{4} B_i \dot{h}_i + \sum_{i=1}^{4} f_{i0} \quad . \end{aligned} \tag{21}$$

But

$$F = M(\ddot{Z}_p - g) \tag{22}$$

where M is the LRV mass and g is the gravity gradient on the surface of the moon. We also know that at rest

$$-Mg = \sum_{i=1}^{4} f_{i0} \quad ,$$

therefore equation (22) can be written as

$$\boxed{F = M\ddot{Z}_0 + \sum_{i=1}^{4} f_{i0} \quad .} \tag{23}$$

In order to formulate the LRV angular motion, we first realize that for small angles

$$T_X \simeq \sum_{i=1}^{4} y_i F_i$$

$$T_Y \simeq - \sum_{i=1}^{4} x_i F_i \quad .$$

Substituting equation (20) into the last two equations yields:

$$\boxed{\begin{aligned} T_X = &- \left(\sum_{i=1}^{4} K_i y_i\right) Z_0 + \left(\sum_{i=1}^{4} K_i x_i y_i\right) \theta_y - \left(\sum_{i=1}^{4} K_i y_i^2\right) \theta_x + \sum_{i=1}^{4} K_i y_i h_i \\ &- \left(\sum_{i=1}^{4} B_i y_i\right) \dot{Z}_0 + \left(\sum_{i=1}^{4} B_i x_i y_i\right) \dot{\theta}_y - \left(\sum_{i=1}^{4} B_i y_i^2\right) \dot{\theta}_x + \sum_{i=1}^{4} B_i y_i \dot{h}_i \end{aligned}} \tag{24}$$

and

$$T_y = \left(\sum_{i=1}^{4} K_i x_i\right) Z_0 - \left(\sum_{i=1}^{4} K_i x_i^2\right) \theta_y + \left(\sum_{i=1}^{4} K_i x_i y_i\right) \theta_x - \sum_{i=1}^{4} K_i x_i h_i$$
$$+ \left(\sum_{i=1}^{4} B_i x_i\right) \dot{Z}_0 - \left(\sum_{i=1}^{4} B_i x_i^2\right) \dot{\theta}_y + \left(\sum_{i=1}^{4} B_i x_i y_i\right) \dot{\theta}_x - \sum_{i=1}^{4} B_i x_i h_i \quad . \quad (25)$$

Note that in equation (24) there is supposed to be an extra term on the right hand side. It is

$$\sum_{i=1}^{4} y_i f_{io} \quad ;$$

however, this is exactly the torque about the x-axis when the LRV is at rest and therefore has to be equal to zero. Similarly, the term

$$\sum_{i=1}^{4} x_i f_{io}$$

in equation (25) vanishes too for the same reason.

The dynamics of the LRV due to torques is expressed by the following set of Euler equations

$$I_x \ddot{\theta}_x + (I_z - I_y) \dot{\theta}_y \dot{\theta}_z = T_x$$
$$I_y \ddot{\theta}_y + (I_x - I_z) \dot{\theta}_z \dot{\theta}_x = T_y$$
$$I_z \ddot{\theta}_z + (I_y - I_x) \dot{\theta}_x \dot{\theta}_y = T_z \quad .$$

Assuming $\dot{\theta}_z = 0$, the first two of the Euler equations are simplified to

$$T_x = I_x \ddot{\theta}_x \quad (26.a)$$

$$T_y = I_y \ddot{\theta}_y \quad (26.b)$$

where I_x and I_y are the LRV moments of inertia about the X_α and Y_α respectively (see Figure 7). Equations (21), (24), (25) and (26) can be written in a single matrix equation as follows

$$\begin{bmatrix} MZ_o \\ I_x\ddot{\theta}_x \\ I_y\ddot{\theta}_y \end{bmatrix} = \begin{bmatrix} -\Sigma K & -\Sigma Ky & \Sigma Kx \\ -\Sigma Ky & -\Sigma Ky^2 & \Sigma Kxy \\ \Sigma Kx & \Sigma Kxy & -\Sigma Kx^2 \end{bmatrix} \begin{bmatrix} Z_o \\ \theta_x \\ \theta_y \end{bmatrix} + \begin{bmatrix} K_1 & K_2 & K_3 & K_4 \\ K_1y_1 & K_2y_2 & K_3y_3 & K_4y_4 \\ -K_1x_1 & -K_2x_2 & -K_3x_3 & -K_4x_4 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$$

$$+ \begin{bmatrix} -\Sigma B & -\Sigma By & \Sigma Bx \\ -\Sigma By & -\Sigma By^2 & \Sigma Bxy \\ \Sigma Bx & \Sigma Bxy & -\Sigma Bx^2 \end{bmatrix} \begin{bmatrix} \dot{Z}_o \\ \dot{\theta}_x \\ \dot{\theta}_y \end{bmatrix} + \begin{bmatrix} B_1 & B_2 & B_3 & B_4 \\ B_1y_1 & B_2y_2 & B_3y_3 & B_4y_4 \\ -B_1x_1 & -B_2x_2 & -B_3y_3 & -B_4x_4 \end{bmatrix} \begin{bmatrix} \dot{h}_1 \\ \dot{h}_2 \\ \dot{h}_3 \\ \dot{h}_4 \end{bmatrix} \qquad (27)$$

where

$$\Sigma K = \sum_{i=1}^{4} K_i$$

$$\Sigma Ky = \sum_{i=1}^{4} K_i y_i$$

$$\Sigma Kxy = \sum_{i=1}^{4} K_i x_i y_i$$

and so on. Laplace transforming equation (27) yields

$$D(s) \begin{bmatrix} Z_o(s) \\ \theta_x(s) \\ \theta_y(s) \end{bmatrix} = N(s) \begin{bmatrix} H_1(s) \\ H_2(s) \\ H_3(s) \\ H_4(s) \end{bmatrix} \qquad (28)$$

where

$$D(s) = \begin{bmatrix} s^2M+s\Sigma B+\Sigma K & s\Sigma By+\Sigma Ky & -s\Sigma Bx-\Sigma Kx \\ s\Sigma By+\Sigma Ky & s^2I_x+s\Sigma By^2+\Sigma Ky^2 & -s\Sigma Bxy-\Sigma Kxy \\ -s\Sigma Bx-\Sigma Kx & -s\Sigma Bxy-\Sigma Kxy & s^2I_y+s\Sigma Bx^2+\Sigma Kx^2 \end{bmatrix}$$

and

$$N(s) = \begin{bmatrix} sB_1+K_1 & sB_2+K_2 & sB_3+K_3 & sB_4+K_4 \\ sB_1y_1+K_1y_1 & sB_2y_2+K_2y_2 & sB_3y_3+K_3y_3 & sB_4y_4+K_4y_4 \\ -sB_1x_1-K_1x_1 & -sB_2x_2-K_2x_2 & -sB_3x_3-K_3x_3 & -sB_4x_4-K_4x_4 \end{bmatrix}$$

We assume that the c.g. of the LRV is at its geometrical center and that the suspensions are identical, thus

$$|x_i| = x_d$$
$$|y_i| = y_d$$

$$K_i = K$$
$$B_i = B \qquad i=1,2,3,4 \quad .$$

Using these relationships the D(s) matrix becomes a diagonal matrix whose inverse is also a diagonal matrix whose elements are the reciprocal of the corresponding elements in the D(s) matrix. Premultiplying equation (28) by the matrix $D^{-1}(s)$ yields

$$\begin{bmatrix} Z_o(s) \\ \theta_x(s) \\ \theta_y(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{Ms^2+4Bs+4K} & \bigcirc & \bigcirc \\ \bigcirc & \frac{1}{I_xs^2+4By_d^2s+4Ky_d^2} & \\ & & \frac{1}{I_ys^2+4Bx_d^2s+4Kx_d^2} \end{bmatrix} .$$

$$\begin{bmatrix} sB+K & sB+K & sB+K & sB+K \\ sBy_d+Ky_d & sBy_d+Ky_d & -(sBy_d+Ky_d) & -(sBy_d+Kx_d) \\ sBx_d+Kx_d & -(sBx_d+Kx_d) & sBx_d+Kx_d & -(sBx_d+Kx_d) \end{bmatrix} \begin{bmatrix} H_1(s) \\ H_2(s) \\ H_3(s) \\ H_4(s) \end{bmatrix} \quad (29)$$

We are interested in the last two equations of this set of three. They can be expressed by the following block diagram:

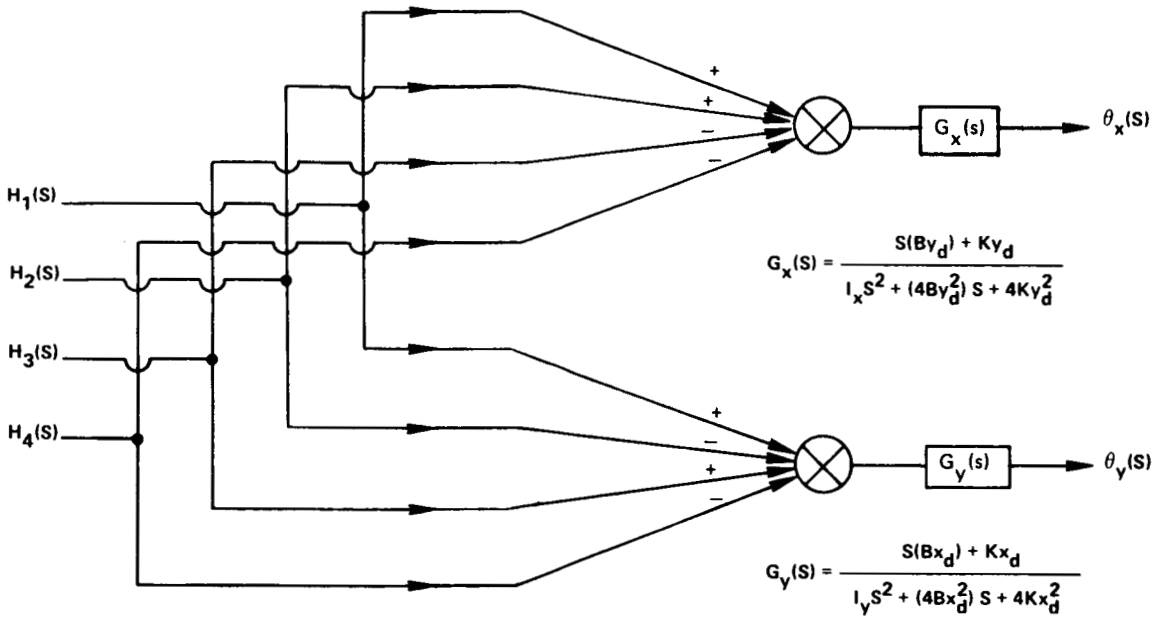


FIGURE 8 - A PARTIAL BLOCK DIAGRAM REPRESENTATION OF THE LRV ROTATIONAL DYNAMICS

The model shown in Figure 8 is simulated in subroutine ANGLES using TRANSIM,[3,4] a transfer function simulation subroutine. The inputs $H_1(s)$, $H_2(s)$, $H_3(s)$ and $H_4(s)$ are obtained from subroutine TERAIN which simulates the elevation of the four LRV wheels during a lunar traverse.

In the model described here we find only θ_x and θ_y. θ_z is the LRV yaw angle. For the deterministic part of θ_z we have the traverse description while for the random part of θ_z we take here the average between θ_x and θ_y.

4.0 THE GENERATION OF THE WHEEL ELEVATIONS

In this subroutine four numbers which represent the wheel elevations are repeatedly generated. Each chain of numbers which represent the elevation of a single wheel satisfies a given PSD function. Moreover, the chains also satisfy certain assumed cross PSD functions between each other.

Given a certain PSD function of some specified lurain, one can generate a chain of numbers which possess this PSD function by passing another chain of numbers, which constitute a white noise, through a suitable shaping filter. Suppose that the PSD function $\Phi_{hlhl}(\omega)$ can be written in the following form

$$\Phi_{hlhl}(\omega) = H(j\omega)\ H(-j\omega) \quad . \tag{30}$$

We recall[6] that if we pass a signal whose PSD function is $\Phi_{xx}(\omega)$ through a filter whose transfer function is W(s), then the PSD of the filtered output signal, y, is given by

$$\Phi_{yy}(\omega) = \Phi_{xx}(\omega)\ W(j\omega)\ W(-j\omega) \quad . \tag{31}$$

Therefore if we pass a signal whose PSD function is 1 through a filter whose transfer function is H(s) then using the rule expressed in equation (31) we obtain the expression shown in equation (30) for $\Phi_{hlhl}(\omega)$. It turns out that 1 is the PSD function of a white noise. From equation (30) we see that in order to find H(jω) which will suit the desired PSD function, one has to separate $\Phi_{hlhl}(\omega)$ into a product of two conjugate functions. This can be achieved if the PSD function plotted on a logarithmic scale is approximated by straight lines and the slope of the i-th line can have only a discrete value which can be any even integer, that is; the i-th slope can have the value $2m_i$, $m_i = 0, \pm 1, \pm 2, \ldots$. Then H(s) is expressed by

$$H(s) = \sqrt{K}\ s^{m_o} \prod_{i=1}^{n} \left(1 + \frac{s}{\Omega_i}\right)^{m_i} \quad . \tag{32}$$

K is the value at ω = 1 rad/sec. of the straight line approximating the lowest frequency region of the PSD function (whether it reaches that point when approximating the PSD function or artificially extended to that point) and m_o is half the slope of this line. Ω_i is the frequency in radians/seconds of the breaking point of the i-th line from the preceeding line, m_i is half of its slope and n is the number of the breaking points. Note that i = 1 corresponds to the first breaking point and the following straight line.

Using this technique one can generate the heights $h_1(t)$ of wheel number one in Figure 7. It is obvious that $h_3(t)$ will be the same sequence of numbers delayed by the time Δt it takes the rear wheel to reach the initial position of the front wheel. About $h_2(t)$, one knows that it has to have the same PSD function as $h_1(t)$; however, there is one important degree of freedom which is the cross correlation between $h_2(t)$ and $h_1(t)$. It is obvious that this freedom exists since for $h_2(t)$ to satisfy the same PSD function which $h_1(t)$ satisfies it is enough that $h_2(t) = h_1(t)$. Yet if $h_2(t) = h_1(t+\hat{T})$ it still satisfies the same PSD function. The latter choice may seem reasonable if $\hat{T}$ corresponds to the time it takes a body to travel from point 2 to point 1 if its speed is equal to the speed of the LRV. This choice however is too rigid; therefore, an assumption concerning the cross PSD functions of h_1 and h_2 is required rather than a relationship between h_1 and h_2 themselves, thus although $h_2(t) \neq h_1(t+T)$ yet we assume that

$$\lim_{t\to\infty} \frac{1}{2t} \int_{-t}^{t} h_1(\lambda)h_2(\lambda+\tau)d\lambda = \lim_{t\to\infty} \frac{1}{2t} \int_{-t}^{t} h_1(\lambda)h_1(\lambda+\tau+T)d\lambda \quad . \quad (33)$$

This equation follows from the assumption that the autocorrelation function of the lurain height along a straight traverse is the same in every direction the wheel traverses on a certain lurain; that is, the lurain autocorrelation function has circular symmetry in the x and y coordinates. Using the symbols for correlation functions, equation (33) is written as

$$\psi_{h1h2}(\tau) = \psi_{h_1h_1}(\tau+\hat{T})$$

and in particular for $\tau = 0$

$$\psi_{h_1h_2}(0) = \psi_{h_1h_1}(\hat{T}) \quad . \tag{34}$$

It is reasonable to assume that

$$\psi_{h_1h_2}(\tau) = a\psi_{h_1h_1}(\tau) \tag{35}$$

where a is a positive number smaller than one which decreases with the increase of $\hat{T}$. For $\tau = 0$ equation (35) becomes

$$\psi_{h_1h_2}(0) = a\psi_{h_1h_1}(0) \quad .$$

From the last equation and equation (34)

$$a\psi_{h_1h_1}(0) = \psi_{h_1h_1}(\hat{T})$$

or

$$a = \frac{\psi_{h_1h_1}(\hat{T})}{\psi_{h_1h_1}(0)} \quad . \tag{36}$$

If it is reasonable to approximate $\Phi_{h_1h_1}(\omega)$ by

$$\Phi_{h_1h_1}(\omega) = \frac{K}{1 + (\frac{\omega}{\Omega})^2}$$

then via an inverse Fourier transform

$$\psi_{h_1h_1}(\tau) = \psi_{h_1h_1}(0) \cdot e^{-|\tau|\Omega}$$

Letting $\tau = \tilde{T}$ and using the result in equation (36) yields

$$a = e^{-\tilde{T}\Omega} \tag{37}$$

Having obtained a, it will now be shown how to generate $h_2(t)$ such that $\Phi_{h_1 h}(\omega) = a\Phi_{h_1 h_1}(\omega)$ and $\Phi_{h_2 h_2}(\omega) = \Phi_{h_1 h_1}(\omega)$.

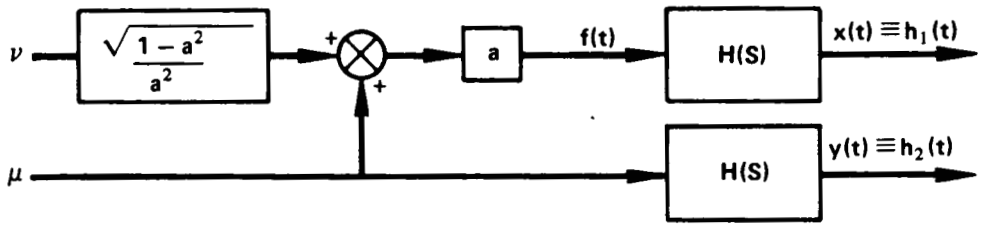


FIGURE 9 - A FILTER TO GENERATE $h_1(t)$ AND $h_2(t)$

Consider Figure 9 where ν and μ are two independent white noise number chains. It is obvious that

$$f = a\left[\mu + \frac{\sqrt{1-a^2}}{a^2}\,\nu\right] \quad .$$

Therefore

$$\psi_{ff}(\tau) = a^2\psi_{\mu\mu}(\tau) + (1-a^2)\psi_{\nu\nu}(\tau) + (1-a^2)\,[\psi_{\mu\nu}(\tau) + \psi_{\nu\mu}(\tau)] \quad .$$

But since ν and μ are white noises then

$$\psi_{\nu\nu}(\tau) = \psi_{\mu\mu}(\tau)$$

and being independent

$$\psi_{\nu\mu}(\tau) = \psi_{\mu\nu}(\tau) = 0 \quad .$$

Therefore

$$\psi_{ff}(\tau) = \psi_{\mu\mu}(\tau)$$

and

$$\Phi_{ff}(\omega) = 1 \tag{38}$$

or in other words f(t) is also a white noise. On the other hand, it can be easily shown that

$$\psi_{f\mu}(\tau) = a\psi_{\mu\mu}(\tau) + \frac{\sqrt{1-a^2}}{a}\psi_{\nu\mu}(\tau)$$

and again since ν and μ are independent

$$\psi_{f\mu}(\tau) = a\psi_{\mu\mu}(\tau) \quad ;$$

therefore,

$$\boxed{\Phi_{f\mu}(\omega) = a \ .} \tag{39}$$

From equation (31) we see that the PSD of the output signals of the filter are

$$\Phi_{xx}(\omega) = \Phi_{ff}(\omega)\ H(-j\omega)H(j\omega) = H(-j\omega)(j\omega)$$

and

$$\Phi_{yy}(\omega) = \Phi_{\mu\mu}(\omega)\ H(-j\omega)H(j\omega) = H(-j\omega)(j\omega)$$

and from equation (30) we realize that

$$\Phi_{xx}(\omega) = \Phi_{yy}(\omega) = \Phi_{h_1h_1}(\omega) \quad . \tag{40}$$

It is well known[5] that for such an arrangement of filters

$$\Phi_{xy}(\omega) = \Phi_{f_\mu}(\omega)\ H(-j\omega)H(j\omega) \quad .$$

Using equation (39)

$$\Phi_{xy}(\omega) = a\ H(-j\omega)H(j\omega)$$

and again, using equation (30) we get

$$\Phi_{xy}(\omega) = a\Phi_{h_1h_1}(\omega) \quad . \tag{41}$$

From equations (40) and (41) we see that x(t) and y(t) satisfy the requirements for the auto and cross-PSD functions of $h_1(t)$ and $h_2(t)$; therefore, they are used as $h_1(t)$ and $h_2(t)$ respectively. In a manner similar to the generation of $h_3(t)$, $h_4(t)$ is equal to $h_2(t + \Delta T)$ where ΔT is the time it takes the rear wheels to reach the position of the front wheels.

In this subroutine the white noise is the output of a random number generator, generating numbers which have a Gaussian distribution with a standard deviation of 1. The PSD function is taken from "MSFC Natural Environment Design Criteria, Criteria Guidelines for Use in Design of Lunar Exploration Vehicles, Exhibit No. 1". The PSD is given in meters^2/cylces/meter) versus cycles/meters. One has to convert the PSD function into meters^2/(radians/second) versus radians/second in order to use it. The net* effect of this conversion on H(s) is that rather than using equation (32), one has to use

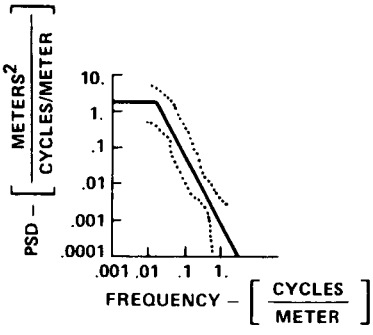
$$H(s) = \sqrt{\frac{k}{4\pi V}}\ s^{m_o} \prod_{i=1}^{n} \left(1 + \frac{s}{\Omega_i V}\right)^{m_i} \tag{42}$$

where V is the LRV velocity in meters/second. Four kinds of PSD functions for four kinds of lurain are given in the above mentioned document. After approximating them by straight lines (as shown in Fig. 10) and application of equation (42) the following filter equations were obtained

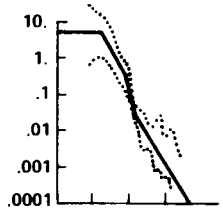
---

*We divide the given PSD function by two since H(s) also simulates the contribution of the negative part of the frequency range.

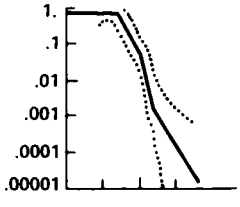
Lurain ID Number	Lurain Description	Filter
1	Smooth Mare	$H(s) = \dfrac{1.265/\sqrt{V \cdot 4\pi}}{1 + \dfrac{s}{0.07536.V}}$
2	Rough Mare	$H(s) = \dfrac{2.239/\sqrt{V4\pi}\,(1 + \dfrac{s}{0.6594V})}{(1 + \dfrac{s}{0.083V})(1 + \dfrac{s}{0.5024V})}$
3	Hummocky Upland	$H(s) = \dfrac{0.944/\sqrt{V4\pi}\,(1 + \dfrac{s}{1.5V})}{(1 + \dfrac{s}{0.17V})(1 + \dfrac{s}{1+0.6V})}$
4	Rough Upland	$H(s) = \dfrac{2.239/\sqrt{V4\pi}\,(1 + \dfrac{s}{0.43V})}{(1 + \dfrac{s}{0.112V})(1 + \dfrac{s}{0.114V})}$



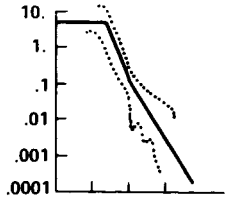
(a) SMOOTH MARE



(b) ROUGH MARE



(c) HUMMOCKY UPLAND



(d) ROUGH UPLAND

FIGURE 10 - THE GIVEN LOWER AND UPPER LIMITS OF THE PSD FUNCTIONS AND THEIR STRAIGHT LINE APPROXIMATION

All the PSD functions of these lurains can be roughly approximated by

$$\Phi_{hh}(\omega) = \frac{K/\sqrt{V \cdot 4\pi}}{1 + (\frac{\omega}{0.07536.V})^2} \quad .$$

Then following equation (37)

$$a = e^{-\tilde{T}(0.07536)V}$$

and using the relationship $\tilde{T} = D/V$, where D is the distance between the two front (or rear) wheels, the last equation becomes

$$a = e^{-(0.07536)D} \quad .$$

The filter shown in Figure 9 is simulated in TERAIN subroutine using TRANSIM and the random number generating routine BARN.[6] As described, the values of $h_3(t)$ and $h_4(t)$ are obtained by delaying $h_1(t)$ and $h_2(t)$ by L/V where L is the distance between the rear to the front wheels.

5.0 TIMING

Consider the block diagram shown in Figure 11.

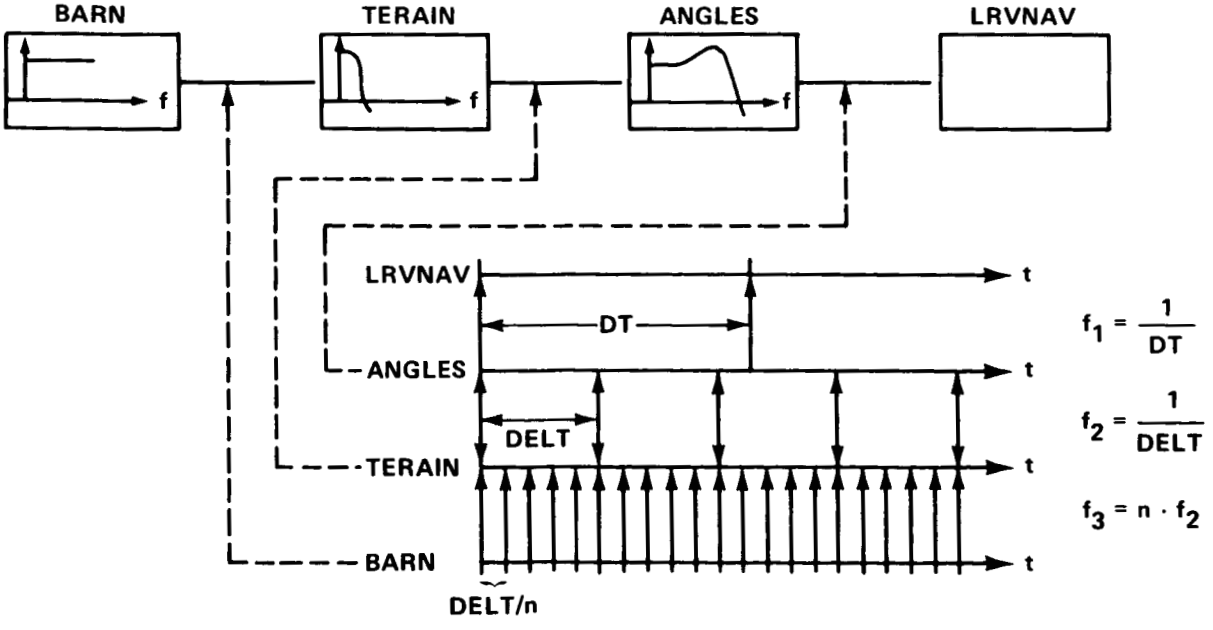


FIGURE 11 - THE FREQUENCY OF SUBROUTINE ENTRANCES

It represents the sequence of data transmission from one routine to the routine it is being called from. In order for the simulation to be efficient from the point of view of time consumption, the execution time should be kept at minimum. Consider the box named ANGLES in Figure 10. In this box the simulation of the LRV dynamics is carried on and hence its responses can be viewed as the responses of some low-pass filters. Therefore it is senseless to ask this subroutine to generate outputs for the main routine, LRVNAV, at a frequency which is much higher than its band-width. A suitable frequency, f_1, of data transmission to the main routine is a frequency which is twice the highest knee frequency of the Bode diagram of the filters in subroutine ANGLES. A criterion similar to this one is used to determine the frequency, f_2, at which ANGLES routine calls TERAIN routine for data. The choice of this frequency is constrained by the fact that there should be an integer number of data points generated in the time $\Delta\tau$ which is defined as the time it takes the rear LRV wheels to reach the position of the front wheels at time t_1 from where the rear wheels were at time t_1. If the distance between the front and rear wheels is L and the LRV speed is V then

$$\Delta\tau = \frac{L}{V} \quad .$$

The reason for this constraint is that the wheel elevation data (these are the data transmitted from TERAIN to ANGLES routine) computed at time t_1 for the front wheel is used $\Delta\tau$ seconds later as the elevations of the rear wheels. The time difference between the instants of data transmission from routine TERAIN to routine ANGLE is chosen therefore as the largest integer part of τ which satisfied the discussed criterion. In general, $f_1 \neq f_2$ or DT $\neq$ DELT (see Figure 11), therefore, LRVNAV usually calls for values from ANGLES at a time instant for which they are not computed. If a set of values has already been computed for a time larger than the time at which the values are called for by LRVNAV (this is possible because the running times for LRVNAV and ANGLES are independent) then an interpolation yields the necessary set. If on the other hand, the last instant for which a set of values were computed is earlier than the time at which a set is needed then ANGLES routine advances in time till its time is at least equal or just greater than the time at which the set is asked for. In the latter case, an interpolation, again, yields the proper set.

While white noise has a constant spectrum at all frequencies, we only require the output of BARN to have a constant spectrum in the pass hand of the filter of TERAIN. The implication of this argument in a continuous system is obvious. In our

case, however, we are dealing with a discrete system where, due to the use of TRANSIM, it is assumed that a straight line connects successive output points of BARN. This way the sampling rate determines not only the cut-off frequency of BARN's ouput but also changes the amplitude of the low frequencies. The sampling frequency of BARN, f_3, was found by a cut and try method. f_3 was chosen as a multiple of f_2 which generates outputs of TERAIN which fit within a reasonable tolerance the given PSD functions of the appropriate lurain.

6.0 INPUT DATA

DT - The time increment of the simulation of the main routine (LRVNAV) in seconds. The program computes DT as was described in section 5, unless it is specified to be otherwise.

TP - The first time at which a printout is desired in minutes.

DTP - The time increment between printouts in minutes.

IROD - If IROD = 0, the program automatically follows the baseline traverse (see Reference 7). If IROD = 1, the program superimposes the random angles generated by subroutine ANGLES on the baseline mission. If IROD = 2, the program follows the traverse imposed by the input data without the random addition from ANGLES. If IROD = 3, the program also does the latter with a superposition of random angles from ANGLES.

LNRT - If other than zero, lunar rotation is taken in account.

OLAMDA - The latitude in degrees of the landing point (important only if the lunar rotation is considered).

MPITCH - Upper limit on the magnitude, in degrees, of the true and computed pitch angle. If it reaches this angle, the program stops.

MAXGAM - The maximum magnitude in degrees of the roll angle to which the astronauts will not respond. When this angle is reached, it is reduced to 5° assuming that the astronauts will feel discomfort and will drive the LRV such that the angle is reduced.

MAXBET - Same for pitch.

SPECDR - When this value of range error in meters is reached, the program stops and prints out.

SPECDB - Same for bearing error in degrees when reached outside a circle whose radius is equal to SPECDR.

IPRDYN - If not zero, the program prints out the coefficients of the transfer functions which are simulated in subroutine ANGLES.

IPRPSD - If not zero, the program does the same in subroutine TERAIN.

YMD - Yaw misalignment error in degrees.

YDR - Yaw drift rate in degrees.

FINALT - The time in minutes for the termination of the run.

DMAX - The maximum distance that the wheels actually travel in kms for a given mission. When this distance is reached, the run is terminated.

IPHOTO - If not equal to zero, the program generates a 4020 plot of the actual traverse.

NPLOT - The number of time increments between points picked for the 4020 plot. (It is set automatically to 20 unless the user specifies a different number.)

M - If you want another run, set M = 1.

IV - For the 0th leg:
If set equal to 1, V = 4.0 km/hr.
If set equal to 2, V = 8.0 km/hr.
If set equal to 3, V = 10.8 km/hr.
If set equal to 4, V = 16.0 km/hr.

LURAIN - For the 0th leg:
Set equal to 1 when LRV traverses on a smooth mare.
Set equal to 2 when LRV traverses on a rough mare.
Set equal to 3 when LRV traverses on a hummocky upland.
Set equal to 4 when LRV traverses on a rough upland.

S - Slip (a number between 0 to 1) during the 0th leg.

W - Azimuth rate of change in radians/hour during the 0th leg.

AZIMUT - The initial azimuth in degrees of the 0th leg.

AZBIAS - The azimuth bias in degrees which has to be added to the true North, as plotted on the traverse map, in order for it to coincide with the map North.

TJUMP(i) - i = 1,2,...,100 - Time in minutes to start the ith traverse leg.

ITABLE(i) - i = 1,2,...,100 - The new speed code for this leg.

LTABLE(i) - i = 1,2,...,100 - The new lurain code for this leg.

STABLE(i) - i = 1,2,...,100 - The new slip for this leg.

WTABLE(i) - i = 1,2,...,100 - The new azimuth rate of change in radians/hour for this leg.

ATABLE(i) - i = 1,2,...,100 - The initial azimuth in degrees for this leg. (If the initial azimuth of this leg is the final azimuth of the preceding leg set ATABLE(i) to a number larger than 360.)

PTABLE(i) - i = 1,2,...,100 - The time in minutes of the duration of the stop at the beginning of this leg.

TMALIN(i) - i = 1,2,...,50 - The time in minutes at which a realignment should take place.

Itzhak Bar-Itzhack

2031-IYB-ajj

I. Y. Bar-Itzhack

Attachments
References
Appendix A

REFERENCES

1. Pio, R. L., "Symbolic Representation of Coordinate Transformation," IEEE Trans. on Aerospace and Navigational Electronics, Vol. ANE-11, pp. 128-134, June 1964.

2. Pio, R. L., "Euler Angle Transformations," IEEE Trans. on Automatic Control, Vol. AC-11, pp. 707-715, October 1966.

3. Andrus, J. F., Integration of Control Equations and the Probelm of Small Time Constants," NASA TN D-3907, Washington, D. C. April 1967.

4. Singers, R. R., "Computer Program Abstract and Write-up for TRANSIM - Transfer Function Simulator Subroutine," Bellcomm Memorandum No. B68-09010, September 5, 1968.

5. G. C. Newton, Jr., L. A. Gould, J. F. Kaiser, "Analytic Design of Linear Feedback Controls", John Wiley & Sons, Inc., New York, 1964.

6. Bellcomm Computer Bulletin, Vol. 7, No. 13, October 21, 1970.

7. Viewgraphs presented at the LRV Preliminary Requirements Review at the Boeing Co., Huntsville, Alabama on December 17, 1969. (The section on the Navigation Subsystem.)

Subject: A Description of the Rover Navigation System Simulation Program - Case 320

From: I. Y. Bar-Itzhack

Distribution List

NASA Headquarters

R. S. Diller/MAT
J. K. Holcomb/MAO
C. M. Lee/MA
A. S. Lyman/MR
B. Milwitzky/MA
W. E. Stoney/MAE

Manned Spacecraft Center

T. J. Blucker/FM4
J. H. Cooper/FC9
C. M. Duke, Jr./CB
G. C. Franklin/CF131
G. D. Griffin/FC
J. B. Irwin/CB
C. Klabosh/CF7
R. H. Kohrs/PD7
G. S. Lunney/FC
J. A. McDivitt/PA
J. Olmsted/CF7
J. M. Peacock/PD7
D. B. Pendley/PA
S. Ritchee/SM
J. E. Saultz/FC9
R. T. Savely/FM4 (3)
H. H. Schmitt/CB
D. R. Scott/CB
J. R. Sevier/PD4
D. K. Slayton/CA
J. L. Smothermon/CF2
H. W. Tindall/FA
J. W. Young/CF
R. G. Zedekar/CF7

Marshall Space Flight Center

J. D. Alter/PM-SAT-LRV
W. A. Armistead/PM-SAT-LRV
J. A. Belew/PM-SAT-LRV
P. H. Broussard, Jr./S&E-ASTR-GC (3)
S. F. Morea/PM-SAT-LRV
W. R. Perry/PM-SAT-LRV
J. M. Sisson/PM-SAT-LRV
E. C. Smith/S&E-ASTR-SG1 (3)
F. W. Wagnon/PD-AP

Boeing Company

S. Allen
R. W. Ekis
R. Goodstein
W. Lerchenmueller
L. J. McMurtry
R. M. O'Brien

Bellcomm

G. M. Anderson
R. A. Bass
A. P. Boysen, Jr.
J. O. Cappellari, Jr.
D. A. DeGraaf
F. El-Baz
D. R. Hagner
J. W. Head
W. G. Heffron
N. W. Hinners
T. B. Hoekstra
W. W. Hough
F. A. LaPiana
M. Liwshitz
J. A. Llewellyn
J. L. Marshall
K. E. Martersteck
J. Z. Menard
P. E. Reynolds
P. F. Sennewald
J. W. Timko
R. A. Troester
R. L. Wagner
All Members Departments 2014, 2031
Department 1024 File
Central File
Library

Distribution List (Contd.)

Abstract Only to

NASA Headquarters

R. A. Petrone/MA

Bellcomm

J. P. Downs
D. P. Ling
M. P. Wilson